DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on February 4, 2010 has been entered.

2. Applicant's amendment dated February 4, 2010, responding to the Final Office action mailed August 4, 2009 provided in the rejection of claims 1, 3-28, 30-36, 38, 40, 42, 44, and 46-55, wherein claims 23 and 32 have been amended, claim 56 has been being newly added.

EXAMINER'S AMENDMENT

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

4. Authorization for this examiner's amendment was given in a telephone interview by Mr. Christopher T. McWhinney (Reg. No. 42,875) on November 29, 2010 to obviate any potential 35 U.S.C 101 and/or 35 U.S.C 112, second paragraph issues and to place the claims in the condition for allowance.

5. In the interest of compact prosecution, the examiner is authorized to further amend the claims 1, 15, 23, 28, and 32 (see Examiner's Amendment below) and to obviate any potential 35 U.S.C 101 and/or 35 U.S.C 112, second paragraph issues.

6. The application has been amended as follows:

IN THE CLAIMS,

Please amend claims 1, 15, 23, 28, and 32 as follows:

1. (Currently amended) A method of processing online transaction messages between a client application and an online transaction processing (OLTP) application server comprising:

receiving a markup language document describing a message, the markup language document being compliant with message rules associated with an ~~online transaction processing (~~OLTP~~)~~ language;

receiving a template for code in a program language, the template describing a data format for the message in a target program language;

automatically generating source code in the program language from the markup language document and the template, the automatically-generated source code depending on both the markup language document and the template, and configured to transform a payload message directly between the target program language data format and an ~~online transaction processing (~~OLTP~~)~~ language message format when compiled and executed, wherein the target program language data format is based on an object-oriented program language and the OLTP language message format is a byte array;

compiling the automatically-generated source code;

a message parser stored on the application server using the compiled automatically-generated source code to transform the payload message directly between the target program language data format and the ~~online transaction processing (~~OLTP~~)~~ language message format.

2. (Canceled).

3. (Previously presented) The method of claim 1, wherein the message rules define markup tags for describing components of the message.

4. (Original) The method of claim 3, wherein the message is a request message

from an application to an OLTP host.

5. (Original) The method of claim 3, wherein the message is a response message from an OLTP host to an application.

6. (Original) The method of claim 3, wherein the markup language is an extensible markup language.

7. (Previously presented) The method of claim 1, wherein the template comprises stylesheet data.

8. (Previously presented) The method of claim 7, further including using a stylesheet parser to automatically generate the source code.

9. (Previously presented) The method of claim 1, further including receiving a plurality of markup language documents for a plurality of message types and generating source code to convert messages of the plurality of message types between the OLTP language and the target language.

10. (Previously presented) The method of claim 1, wherein the target language is an object-oriented program language.

11. (Original) The method of claim 10, further including:

receiving one or more name-value pairs for the message according to the object- oriented program language;

generating a byte array according to the OLTP language; and

transmitting the byte array to an application.

12. (Original) The method of claim 10, further including:

receiving a byte array for the message according to the OLTP language; generating one or more name-value pairs according to the object-oriented program language; and

transmitting the name-value pairs to an application.

13. (Original) The method of claim 10, wherein the object-oriented program language is a Java language.

14. (Original) The method of claim 1, wherein the OLTP language is a gaming system language.

15. (Currently amended) A method of generating source code for processing online transaction messages between a client application and an online transaction processing (OLTP) application server, comprising:

receiving a markup language document, the markup language document being compliant with message rules associated with an ~~online transaction processing (~~OLTP~~)~~ language;

receiving a template for code in a program language, the template describing a data format in a target program language for a message; and

automatically generating a source code based on the markup language document and the template, the source code depending on both the markup language document and the template, and enabling transformation of a payload message directly between the target program language data format and an OLTP language message format, wherein the target program language data format is based on an object-oriented program language and the OLTP language message format is a byte array.

16. (Original) The method of claim 15, wherein the message rules define markup tags for describing components of the message.

17. (Original) The method of claim 16, wherein the message is a request message from an application to an OLTP host.

18. (Original) The method of claim 16, wherein the message is a response message from an OLTP host to an application.

19. (Original) The method of claim 16, wherein the markup language is an extensible markup language.

20. (Previously presented) The method of claim 15, wherein the template comprises stylesheet data.

21. (Previously presented) The method of claim 20, further including using a stylesheet parser to automatically generate the source.

22. (Previously presented) The method of claim 15, further including receiving a plurality of markup language documents for a plurality of message types and generating source code to convert the plurality of message types between the OLTP language and the target language.

23. (Currently amended) A method of processing online transaction messages between a client application and an online transaction processing (OLTP) application server comprising:

receiving one or more extensible markup language (XML) documents, the XML documents being compliant with message rules associated with an ~~online transaction processing (~~OLTP~~)~~ language, the OLTP language being a gaming system language and the message rules defining markup tags for describing components of ~~the~~ a message;

automatically compiling the XML documents and generating ~~into~~ source code in an object-oriented program language based on the XML documents and stylesheet data, the stylesheet describing a data format for the message in a target program language, the source code configured, when compiled and executed, to transform a message directly between the target program language data format and an OLTP language message format, wherein the target program language data format is based on an object-oriented program language and the OLTP language message format is a byte array;

compiling the source code;

a message parser stored on the application server using the compiled source code to transform a payload message directly between the target language data format and the OLTP language message format; and repeating the receiving and compiling of XML documents and the compiling of source code for a plurality of message types.

24. (Original) The method of claim 23, further including using a stylesheet parser to compile the XML documents.

25. (Original) The method of claim 23, further including: receiving one or more name-value pairs for the message according to the program language; generating a byte array according to the OLTP language; and transmitting the byte array to an application.

26. (Original) The method of claim 23, further including: receiving a byte array for the message according to the OLTP language; generating one or more name-value pairs according to the program language; and transmitting the name-value pairs to an application.

27. (Original) The method of claim 23, wherein the object-oriented program language is a Java language.

28. (Currently amended) An ~~A message processing~~ apparatus for processing online transaction messages between a client application and an online transaction processing (OLTP) application server, comprising:

an application server including a development module stored on the application server, the development module configured to receive a markup language document and a template for code in a program language, the template describing a message data format in a target program language, and configured to automatically generate a source code in the program language, the source code depending on both the markup language document and the template, the markup language document to be compliant with message rules associated with an ~~the~~ OLTP language, and the source code configured to enable direct transformation of a message between the target program language message data format and an OLTP language message format, wherein the target program language data format is based on an object-oriented program language and the OLTP language message format is a byte array;

a program compiler ~~stored on the application server, the program compiler~~ configured to compile the source code; and

a message parser stored on the application server, the message parser configured to transform a payload message directly between the target program language message data format and the ~~online transaction processing (~~OLTP~~)~~ language message format using the compiled source code.

29. (Canceled).

30. (Previously presented) The system of claim 28, wherein the message rules define markup tags for describing components of the message.

31. (Previously presented) The system of claim 28, wherein the development module further includes a stylesheet parser, the development module to use the stylesheet parser to automatically generate the source code based on stylesheet data.

32. (Currently amended) A machine readable medium comprising a set of stored instructions capable of being executed by a processor to perform a method comprising the steps of:

~~receive~~ receiving a markup language document, the markup language document being compliant with message rules associated with an online transaction processing (OLTP) language;

~~receive~~ receiving a template for code in a program language, the template describing a data format in a target program language relating to a message; and

automatically ~~generate~~ generating a source code based on the markup language document and the template, the source code depending on both the markup language document and the template, and configured to transform a payload message directly between the target program language data format and an OLTP language message format [[,]] when the source code is compiled and executed, wherein the target program language data format is based on an object-oriented program language and the OLTP language message format is a byte array ~~when the source code is compiled and executed~~.

33. (Original) The medium of claim 32, wherein the message rules define markup tags for describing components of the message.

34. (Previously presented) The medium of claim 32, wherein the template comprises stylesheet data.

35. (Previously presented) The medium of claim 32, wherein the instructions are further capable of being executed to receive a plurality of markup language documents for a plurality of message types and to generate source code for converting the plurality of message types between the OLTP language message format and the target language message format.

36. (Previously presented) The method of claim 1, further comprising:

providing an online gaming system by an OTLP host which uses the OLTP language, wherein the online gaming system processes wagering-related transactions; and

providing an application at a consumer location which uses the target program language, wherein the application provides access to the wagering-related transactions for the consumer,

wherein the payload message is part of at least one of the wagering-related transactions and comprises at least one of: a request to place an online wager, or a response to the request to place the online wager.

37. (Canceled).

38. (Previously presented) The method of claim 15, further comprising:

providing an online gaming system by an OTLP host which uses the OLTP language, wherein the online gaming system processes wagering-related transactions; and

providing an application at a consumer location which uses the target program language, wherein the application provides access to the wagering-related transactions for the consumer,

wherein the payload message is part of at least one of the wagering-related transactions and comprises at least one of: a request to place an online wager, or a response to the request to place the online wager.

39. (Canceled).

40. (Previously presented) The method of claim 23, further comprising:

providing an online gaming system by an OTLP host which uses the OLTP language, wherein the online gaming system processes wagering-related transactions; and

providing an application at a consumer location which uses the target program language, wherein the application provides access to the wagering-related transactions for the consumer,

wherein the payload message is part of at least one of the wagering-related transactions and comprises at least one of: a request to place an online wager, or a response to the request to place the online wager.

41. (Canceled).

42. (Previously presented) The method of claim 28, further comprising:

providing an online gaming system by an OTLP host which uses the OLTP language, wherein the online gaming system processes wagering-related transactions; and

providing an application at a consumer location which uses the target program language, wherein the application provides access to the wagering-related transactions for the consumer, wherein the payload message is part of at least one of the wagering-related transactions and comprises at least one of: a request to place an online wager, or a response to the request to place the online wager.

43. (Canceled).

44. (Previously presented) The machine readable medium of claim 32, wherein the set of stored instructions is capable of being executed by the processor to further:

provide an online gaming system by an OTLP host which uses the OLTP language, wherein the online gaming system processes wagering-related transactions; and

provide an application at a consumer location which uses the target program language, wherein the application provides access to the wagering-related transactions for the consumer,

wherein the payload message is part of at least one of the wagering-related transactions and comprises at least one of: a request to place an online wager, or a response to the request to place the online wager.

45. (Canceled).

46. (Previously presented) The method of claim 1, wherein the program language is the target program language.

47. (Previously presented) The method of claim 15, wherein the program language is the target program language.

48. (Previously presented) The method of claim 23, wherein the program language is the target program language.

49. (Previously presented) The apparatus of claim 28, wherein the program

language is the target program language.

50. (Previously presented) The machine readable medium of claim 32, wherein the program language is the target program language.

51. (Previously presented) The method of claim 1, wherein the automatically-generated source code is further configured to transform the first payload message directly from the target program language data format to the online transaction processing (OLTP) language message format, and to transform a second payload message directly from the OLTP language message format to the target program language data format.

52. (Previously presented) The method of claim 15, wherein the source code is further configured to enable transformation of the payload message directly from the target program language data format to the OLTP language message format, and to enable transformation of a second payload message directly from the OLTP language message format to the target program language data format.

53. (Previously presented) The method of claim 23, wherein the source code is further configured, when compiled and executed, to transform the message directly from the target program language data format to the OLTP language message format, and to transform a second message directly from the OLTP language message format to the target program language data format.

54. (Previously presented) The apparatus of claim 28, wherein the source code is further configured to enable direct transformation of the message from the target program language message data format to the OLTP language message format, and to enable direct transformation of a second message from the OLTP language message format to the target program language message data format.

55. (Previously presented) The machine readable medium of claim 32, wherein the source code is further configured to transform the payload message directly

from the target program language data format to the OLTP language message format, and to transform a second payload message directly from the OLTP language message format to the target program language data format.

56. (Previously presented) The method of claim 48, further comprising:

using a stylesheet parser to compile the XML documents;

receiving one or more name-value pairs for the message according to the program language; generating a byte array according to the OLTP language; and transmitting the byte array to an application;

receiving a byte array for the message according to the OLTP language; generating one or more name-value pairs according to the program language; and transmitting the name-value pairs to an application; and

wherein the object-oriented program language is a Java language.

- **END OF AMENDMENT –**

Allowable Subject Matter

7. Claims 1, 3-28, 30-36, 38, 40, 42, 44, and 46-56 (renumbered as 1-49) are allowed.

8. The following is an examiner's statement of reasons for allowance:

The cited prior arts of record (*Zorc, Koch,* and *Lai*) taken alone or in combination does not disclose and/or fairly suggest a method, an apparatus and/or a software product for

"… processing online transaction messages between a client application and an online transaction processing (OLTP) application server comprising:

receiving a markup language document describing a message, the markup language document being compliant with message rules associated with an OLTP language;

receiving a template for code in a program language … describing a data format for the message in a target program language;

automatically generating source code in the program language from the markup language document and the template … configured to transform a payload message directly between the target program language data format and an OLTP language message format … wherein the target program language data format is based on an object-oriented program language and the OLTP language message format is a byte array;

compiling the automatically-generated source code;

a message parser … to transform the payload message directly between the target program language data format and the OLTP language message format .”, as recited in such manners in independent claim 1 and similarly recited in independent claim 15, 23, 28, and 32.

9. Claims (3-14, 36, 46, 51), (16-22, 38, 47, 52), (24-27, 40, 48, 53, 56), (30-31, 42, 49, 54), and (33-35, 44, 50, 55) are considered allowable by virtue of their dependence on allowable independent claims 1, 15, 23, 28, and 32 respectively.

10. Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled “Comments on Statement of Reasons for Allowance.”

Conclusion

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Ben C. Wang whose telephone number is (571) 270-1240. The examiner can normally be reached on 8:00-5:30 (EST/EDT), Monday through Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system.

Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Ben C Wang/
Examiner, Art Unit 2192

/Michael J. Yigdall/
Primary Examiner, Art Unit 2192